PEMBUATAN APLIKASI INFORMASI NILAI SISWA MENGGUNAKAN KONSEP MODEL VIEW CONTROLLER (MVC)

Dea Ayu Nurul Qulbi[1], Yuniana Cahyaningrum[2]

Pendidikan Teknologi Informasi, IKIP PGRI Bojonegoro

[1]Email: deaayunur23@gmail.com

[2]Email: yuniana@ikippgribojonegoro.ac.id

ABSTRAK

Berkembangnya teknologi saat ini semakin luas dalam bidang apapun, salah satunya dibidang teknologi pemrograman seperti konsep Model View Controller (MVC). Model View Controller pertama sekali dipublikasikan oleh peneliti XEROX PARC yang bekerja dalam pembuatan bahasa pemrograman Smalltalk sekitar tahun 1970-1980, hingga sekarang software design pattern ini banyak dipergunakan dalam pengembangan sebuah aplikasi maupun situs web. Model View Controller adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (Model) dari tampilan (View) dan cara bagaimana memprosesnya (Controller). Dalam implementasinya kebanyakan framework dalam aplikasi website adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web. (Hamdi, Muhammad.2019). Metodologi penelitan yang digunakan dalam artikel ini adalah metode kualitatif melalui studi pustaka. Metode kualitatif adalah prosedur penelitian yang menghasilkan data deskriptif berupa kata-kata, catatan-catatan yang berhubungan dengan arti, nilai serta pengertian.

Kata Kunci: *MVC, Pengertian MVC, Manfaat MVC, Cara Kerja MVC*

ABSTRACT

The development of technology is currently increasingly widespread in any field, one of which is in the field of programming technologysuch as the Model View Controller (MVC) concept. The Model View Controller was first published by XEROC PARC researchers who worked on the Smalltalk Programming language around 1970-1980, until now this software design pattern is widely use in developing an application or website. Model View Controller is a method for creating an application by separating the data (Model), from the view (View), and how to process it (Controller). In its implementation, most frameworks in website applications are based on the MVC architecture. MVC separates application development based on the main components that build an application such as data manipulation, user interface, and parts that become controls in a web application. (Hamdi, Muhammad.2019). The research methodology

used in this article is a qualitative method through literature study. Qualitative methods are research procedures that produce descriptive data in the form of words, notes related to meaning, value and understanding.

Keywords: *MVC, Definition of MVC, Benefits of MVC, How MVC works*

PENDAHULUAN

Berkembangnya teknologi saat ini semakin luas dalam bidang apapun, seperti teknologi dibidang pendidikan, teknologi sistem Informasi, teknologi pemrograman dan sebagainya. Contoh di bidang pendidikan seperti adanya aplikasi zoom/Gmeet, Media sosial learning dan sebagainya. Contoh si bidang teknologi informasi seperti Handphone, televisi, PC, dan sebagainya. Sedangkan contoh teknologi di bidang pemrograman seperti adanya konsep sistem pemrograman Model View Controller (MVC).

Model View Controller adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (Model) dari tampilan (View) dan cara bagaimana memprosesnya (Controller). Dalam implementasinya kebanyakan framework dalam aplikasi website adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web. (Hamdi, Muhammad.2019).

Dengan konsep mvc ini bisa mempermudah dalam mengerjakan suatu projek, salah satunya projek pembuatan aplikasi sistem informasi. Perkembangan sistem informasi yang pesat saat ini maka penyajian informasi secara efektif semakin dibutuhkan. Penyajian informasi seharusnya sudah beralih dari sistem lama yaitu pencatatan manual menjadi system komputerisasi. Demikian halnya pencatatan data nilai siswa yang akan di bahas dalam artikel kali ini.

Di sekolah para siswa setiap semesternya pasti ada yang namanya penilaian mata pelajaran. Penilaian mata pelajaran biasanya lewat beberapa penilaian harian, penilaian tengah semester, dan akhir semester. Penilaian tersebut akan di masukkan dalam buku rapor. Sebelum di masukkan ke rapor, setiap guru akan mencatatnya dahulu di kertas nilai siswa lalu di kelola lagi di Excel untuk di masukkan ke rapor, dengan kata lain dua kali kerja. Untuk mempermudah pencatatan maka penulis berniat membuat sebuah aplikasi untuk pencatatan nilai siswa. Tujuan dibuatnya aplikasi ini untuk mempermudah pencatatan nilai (tidak kerja dua kali), dan menghindari kesalahan pencatatan (human error).

KAJIAN PUSTAKA

Model View Controller pertama sekali dipublikasikan oleh peneliti XEROX PARC yang bekerja dalam pembuatan bahasa pemrograman Smalltalk sekitar tahun 1970-1980, hingga sekarang software design pattern ini banyak dipergunakan dalam pengembangan sebuah aplikasi maupun situs web. Design pattern ini sekarang banyak dimanfaatkan pada PHP Framework populer di dunia seperti Laravel, Symfony, Code Igniter, maupun Yii. MVC memisahkan

program dalam tiga bagian terpisah yang saling terhubung sesuai dengan namanya, yaitu: Model, View, dan Controller.

MVC atau Model View Controller adalah sebuah cara dalam membuat aplikasi atau website dengan Model yang mana model ini berisi beberapa data yang akan di gunakan membuat projek, dan View berisi tampilan projek tersebut, dan Controller berisi perintah-perintah yang memiliki fungsi (function) dalam menghubungkan model dan view tersebut.

Jadi bisa disimpulkan MVC ini seperti:

1. Model, merupakan bagian untuk pengelolaan pada database
2. View, merupakan bagian yang menampilkan desain tampilan dan juga informasi kepada user atau pengguna (end user)
3. Controller, merupakan bagian yang memberikan fungsi dalam menghubungkan model dan view pada setiap proses request dari user.

Dengan konsep MVC, pengembangan sebuah website atau aplikasi lebih mudah di kerjakan karena pembuatan website tersebut terdiri dari masing-masing bagian yang terpisah sehingga tim developer dapat lebih fokus ke salah satu bagiannya saja dari model, view, dan controller.

Cara kerja MVC adalah pada bagian View melakukan permintaan informasi agar dapat ditampilkan kepada pengguna. Permintaan tersebut kemudian diterima oleh Controller dan dikirimkan ke bagian Model untuk diproses. Di bagian Model, informasi tersebut akan diolah dan dicari data informasinya ke dalam database yang dimiliki. Setelahnya, Model akan memberikan kembali pada controller untuk ditampilkan hasilnya pada View. Controller mengambil hasil olahan yang dilakukan pada bagian model dan menatanya pada bagian View.

Berikut beberapa manfaat menggunakan framework MVC dalam pembuatan website:

1. Proses Pengembangan Website Menjadi Lebih Efisien

   Konsep MVC membantu membuat proses pengembangan website menjadi lebih cepat. Hal ini karena MVC membagi bagian Model, View dan Controller. Jadi jika dalam pembuatan website ini terbagi dalam tim, maka pengerjaannya akan terbagi seperti ini: Bagian Model dan Controller bisa dikerjakan oleh Tim Back End Developer. Bagian View bisa dikerjakan oleh Tim Front End Developer dan Tim UI/UX. Jadi, tim UI/UX dapat menyelesaikan perancangan desain pada halaman depan, tim Back End dan Front End dapat memulai membuat kode pemrograman. Tim UI/UX juga dapat beralih ke perancangan halaman desain lain misalnya halaman produk.

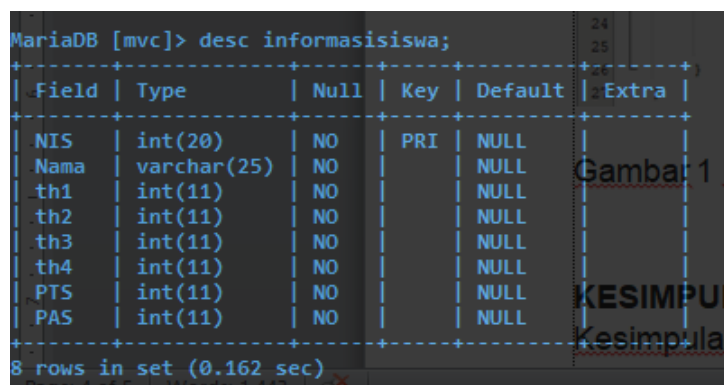
2. Pengujian Menjadi Lebih Mudah

   Sangat diperlukan sebuah tahapan pengujian yang harus dilakukan agar dapat memastikan website dapat berjalan dengan baik. Jadi, proses pengujian dapat dilakukan per bagian yang telah siap dan tidak perlu menunggu seluruh situs web selesai.

3. Error atau Bug Lebih Cepat dan Mudah Ditangani

Metode MVC dapat membuat developer lebih fokus pada bagian pengembangannya masing-masing. Jadi mereka dapat menemukan bug dengan cepat dan dapat segera memperbaikinya. Contohnya, sebuah fungsi Simpan pada website rapor online tidak dapat berjalan dengan baik, maka kemungkinan permasalahan yang bisa terjadi adalah pada datanya atau tombolnya. Jadi, jika ternyata errornya adalah kesalahan pada datanya yaitu di bagian Model, maka bagian front end dapat langsung memperbaiki error tersebut.

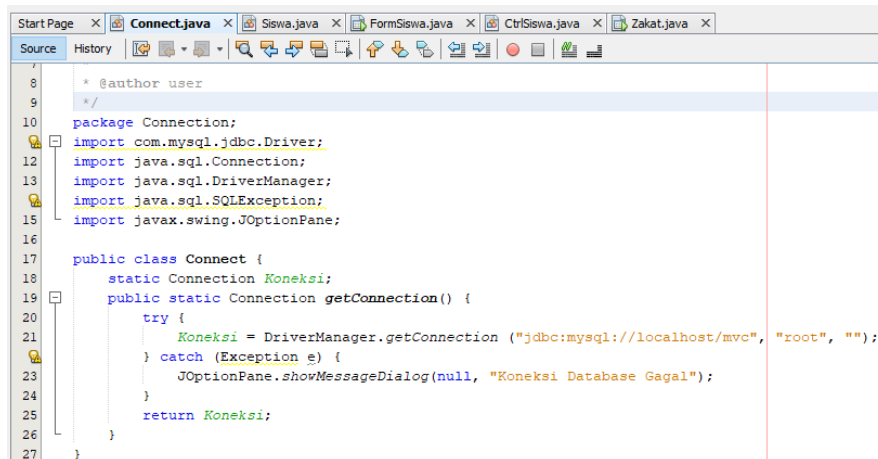
4. Pemeliharaan atau Maintenance Menjadi Lebih Mudah

   Selain kemudahan-kemudahan di atas, konsep MVC ini juga dapat memudahkan dalam menyusun script menjadi lebih terstruktur dan rapi. Jika suatu saat ingin melakukan pengembangan atau maintenance website tersebut, tim developer akan mudah melakukannya. Contohnya, jika ingin menambahkan fitur "tabel absensi" pada website Rapor Online, maka tim developer dapat dengan mudah menemukan kode script yang dapat ditambahkan atau diperbarui.

METODE

Metodologi penelitan yang digunakan dalam artikel ini adalah metode kualitatif melalui studi pustaka. Metode kualitatif adalah prosedur penelitian yang menghasilkan data deskriptif berupa kata-kata, catatan-catatan yang berhubungan dengan arti, nilai serta pengertian. Sumber data dalam penelitian ini, terdiri dari beberapa sumber seperti website dan juga artikel. Dari sumber tersebut penulis juga dapat membuat sebuah aplikasi sistem informasi Nilai Siswa Sederhana dengan bantuan aplikasi seperti XAMPP untuk menghubungkan ke Database, dan juga Netbeans untuk menuliskan kode program.

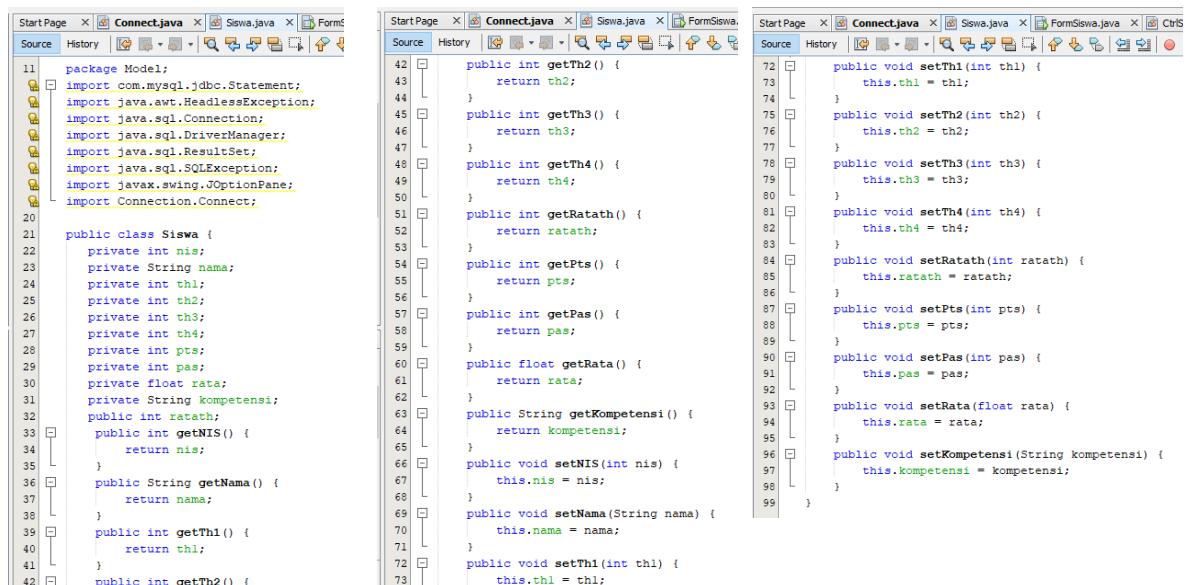
PEMBAHASAN

Agar bisa lebih paham dengan kosnep MVC, berikut adalah contoh implementasi konsep MVC dengan kode program Java menggunakan aplikasi Netbeans, berikut langkah-langkahnya:

1. Buatlah sebuah database dahulu di localhost atau di Command prompt

```
MariaDB [mvc]> desc informasisiswa;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| NIS   | int(20)     | NO   | PRI | NULL    |       |
| Nama  | varchar(25) | NO   |     | NULL    |       |
| th1   | int(11)     | NO   |     | NULL    |       |
| th2   | int(11)     | NO   |     | NULL    |       |
| th3   | int(11)     | NO   |     | NULL    |       |
| th4   | int(11)     | NO   |     | NULL    |       |
| PTS   | int(11)     | NO   |     | NULL    |       |
| PAS   | int(11)     | NO   |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
8 rows in set (0.162 sec)
```

Gambar 1 Database

Gambar diatas merupapkan database yang akan digunakan di program Informasi Nilai Siswa. Database tersebut dibuat di command prompt menggunakan nama *database mvc* dengan nama *tabel informasisiswa*.

2. Buat project di Netbeans dan koneksikan ke database

```
 * @author user
 */
package Connection;
import com.mysql.jdbc.Driver;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.JOptionPane;

public class Connect {
    static Connection Koneksi;
    public static Connection getConnection() {
        try {
            Koneksi = DriverManager.getConnection ("jdbc:mysql://localhost/mvc", "root", "");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Koneksi Database Gagal");
        }
        return Koneksi;
    }
}
```

Gambar 2 Koding untuk koneksi ke database

Buatlah file di Netbeans dengan nama *mvc*, lalu di file mvc tersebut buatlah sebuah *package Connection* lalu Class-nya beri nama *Connect*. Di *Class Connect()* isi dengan kode diatas.

3. Buat Package Model
   Buat package lagi dengan nama Model. Di dalam package Model buatlah *Class Siswa()* yang nantinya akan diisi data-data

Gambar 3 *Class Siswa()* (Model)

Pada *Class Siswa()* berisi data-data yang akan digunakan dalam membuat program Informasi Nilai Siswa seperti pada gambar diatas.

4. Buat package View

   Pada package View buatlah JFrame Form dengan nama *FormSiswa()* untuk membuat desain atau tampilan programnya

```java
package View;
import Controller.CtrlSiswa;
import javafx.application.Application;
import javax.swing.JTextArea;
import javax.swing.JTextField;
    public class FormSiswa extends javax.swing.JFrame {
    private CtrlSiswa ctrl;
    public FormSiswa() {
        initComponents();
        ctrl = new CtrlSiswa(this);
    }
    public JTextField gettxtnis() {
        return txtnis;
    }
    public JTextField gettxtnama() {
        return txtnama;
    }
    public JTextField gettxtth1() {
        return txtth1;
    }
    public JTextField gettxtth2() {
        return txtth2;
    }
    public JTextField gettxtth3() {
        return txtth3;
    }
    public JTextField gettxtth4() {
        return txtth4;
    }
    public JTextField gettxtpts() {
        return txtpts;
    }
```

```java
    public JTextField gettxtth2() {
        return txtth2;
    }
    public JTextField gettxtth3() {
        return txtth3;
    }
    public JTextField gettxtth4() {
        return txtth4;
    }
    public JTextField gettxtpts() {
        return txtpts;
    }
    public JTextField gettxtpas() {
        return txtpas;
    }
    public JTextArea gettxthasil() {
        return txthasil;
    }
```

Gambar 4.1 Koding deklarasi variabel di *Class FormSiswa()*.

Gambar 4.2 Interface

5. Buat package Controller

```
Start Page | Connect.java | Siswa.java | FormSiswa.java | CtrlSiswa.java | Zakat.java
Source History

10  package Controller;
11  import Connection.Connect;
12  import Model.Siswa;
13  import View.FormSiswa;
14  import com.mysql.jdbc.Statement;
    import java.sql.Connection;
    import java.sql.PreparedStatement;
17  import javax.swing.JOptionPane;
18
19  public class CtrlSiswa {
20      private Siswa sis;
        private FormSiswa tampil;
22
23      public CtrlSiswa (FormSiswa tampil) {
24          this.tampil = tampil;
25      }
26
27      public void simpan () {
28          try {
                Statement statement = (Statement) Connect.getConnection().createStatement();
30              statement.executeUpdate ("insert into informasisiswa VALUES ('"+sis.getNIS()+"', '"+sis.getNama()+"', '"
31              +sis.getTh1()+"', '"+sis.getTh2()+"', '"+sis.getTh3()+"', '"+sis.getTh4()+"', '"+sis.getPts()+"', '"
32              +sis.getPas()+"');");
33              statement.close ();
34              JOptionPane.showMessageDialog (null, "Data Berhasil Disimpan");
            } catch (Exception t) {
36              JOptionPane.showMessageDialog (null, "Data Gagal Disimpan");
37          }
38          }
```

Gambar 5.1 *Class CtrlSiswa()*, Import, Deklarasi variable, Fungsi Simpan.

```
39
40      public void proses() {
41          sis = new Siswa();
42          sis.setNIS(Integer.parseInt(tampil.gettxtnis().getText()));
43          sis.setNama(tampil.gettxtnama().getText());
44          sis.setThl(Integer.parseInt(tampil.gettxtthl().getText()));
45          sis.setTh2(Integer.parseInt(tampil.gettxtth2().getText()));
46          sis.setTh3(Integer.parseInt(tampil.gettxtth3().getText()));
47          sis.setTh4(Integer.parseInt(tampil.gettxtth4().getText()));
48          sis.setPts(Integer.parseInt(tampil.gettxtpts().getText()));
49          sis.setPas(Integer.parseInt(tampil.gettxtpas().getText()));
50          sis.ratath = ((sis.getThl() + sis.getTh2() + sis.getTh3() + sis.getTh4())/4);
51          sis.setRata((sis.getRatath() + sis.getPts() + sis.getPas())/3);
52          if (sis.getRata() >= 75) {
53              sis.setKompetensi ("KOMPETEN");
54          } else {
55              sis.setKompetensi("TIDAK KOMPETEN");
56          }
57          tampil.gettxthasil().setText("INFORMASI NILAI SISWA\n"
58                  +"NIM : "+sis.getNIS()+"\n"
59                  +"Nama : "+sis.getNama()+"\n"
60                  +"Tugas Harian : "+sis.getRatath()+"\n"
61                  +"Penilaian Tengah Semester : "+sis.getPts()+"\n"
62                  +"Penilaian Akhir Semester : "+sis.getPas()+"\n"
63                  +"Rata-rata : "+sis.getRata()+"\n"
64                  +"Anda Dinyatakan "+sis.getKompetensi()+"\n");
65      }
66
67          public void reset () {
68              tampil.gettxtnis().setText("");
69              tampil.gettxtnama().setText("");
70              tampil.gettxtthl().setText("");
```

Gambar 5.2 *Class CtrlSiswa()*, Fungsi Proses, Fungsi Reset.

```
66
67          public void reset () {
68              tampil.gettxtnis().setText("");
69              tampil.gettxtnama().setText("");
70              tampil.gettxtthl().setText("");
71              tampil.gettxtth2().setText("");
72              tampil.gettxtth3().setText("");
73              tampil.gettxtth4().setText("");
74              tampil.gettxtpts().setText("");
75              tampil.gettxtpas().setText("");
76              tampil.gettxthasil().setText("");
77          }
78
79          public void hapus() {
80          sis.setNIS(Integer.parseInt(tampil.gettxtnis().getText()));
81          sis.setNama(tampil.gettxtnama().getText());
82          sis.setThl(Integer.parseInt(tampil.gettxtthl().getText()));
83          sis.setTh2(Integer.parseInt(tampil.gettxtth2().getText()));
84          sis.setTh3(Integer.parseInt(tampil.gettxtth3().getText()));
85          sis.setTh4(Integer.parseInt(tampil.gettxtth4().getText()));
86          sis.setPts(Integer.parseInt(tampil.gettxtpts().getText()));
87          sis.setPas(Integer.parseInt(tampil.gettxtpas().getText()));
88              try {
                Statement statement = (Statement) Connect.getConnection().createStatement();
90              statement.executeUpdate("DELETE FROM informasisiswa WHERE NIS='"+sis.getNIS()+"'");
91              statement.close ();
92              JOptionPane.showMessageDialog (null, "Data Berhasil Di Hapus");
            } catch (Exception t) {
94              JOptionPane.showMessageDialog (null, "Data Gagal Di Hapus");
95          }
96          }
97      }
```

Gambar 5.3 *Class CtrlSiswa()*, Fungsi Reset, Fungsi Hapus.

Pada *Class CtrlSiswa()* berisi semua perintah atau fungsi untuk menjalankan View dengan mengambil data dari Model. Kemudian semua fungsi tersebut akan dipanggil di View sesuai fungsinya seperti pada gambar di bawah ini.

Gambar 5.4 *Class FormSiswa()*, Pemanggilan Fungsi di *Class CtrlSiswa()* ke *Class FormSiswa()*.

KESIMPULAN

Kesimpulan yang di dapat dari pembahasan di atas adalah:

1. Model, berisi data-data yang akan digunakan dalam pembuatan program.
2. View, berisi desain tampilan dan juga informasi kepada user atau pengguna (end user).
3. Controller, berisi fungsi dalam menghubungkan model dan view pada setiap proses request dari user.
4. MVC memudahkan perawatan sebuah program, karena pembuatan program tersebut terdiri dari masing-masing bagian yang terpisah sehingga tim developer dapat lebih fokus ke salah satu bagiannya saja dari model, view, dan controller.

DAFTAR PUSTAKA

Anonim.2022.***APA ITU MODEL VIEW CONTROLLER MVC***:https://lp2mp.uma.ac.id/apa-itu-model-view-controller-mvc/, diakses pada tanggal 02 Juni 2023

Coding, Sobat.2021.***MVC SEDERHANA PENGELOLAAN NILAI SISWA (JAVA)***:https://youtu.be/Uw8PDBiKT4w, diakses pada tanggal 25 Mei 2023

Hamdi, Muhammad.2019.***KONSEP MVC PADA PEMROGRAMAN***:http://lea.si.fti.unand.ac.id/2019/04/konsep-mvc-pada-pemrograman/, diakses pada tanggal 02 Juni 2023

Sew.2016.***SISTEM PEMROGRAMAN MODEL VIEW CONTROLLER MVC***:https://informatika.uc.ac.id/2016/12/sistem-pemrograman-model-view-controller-mvc/#, diakses pada tanggal 02 Juni 2023